FFHS
Fernfachhochschule Schweiz
Zürich | Basel | Bern | Brig

Mitglied der SUPSI

Janick Rüegger

Hashed Timelock Contracts: An Incentive Analysis

Agent Synthesis and Rational Verification for fully-decentralized Atomic Cross-Chain Swaps

Bachelor's Thesis

Departement Informatik
Fernfachhochschule Schweiz (FFHS)

Submitted to

Dr. Oliver Kamin

Supervision

Dr. Ilir Fetai & Dr. Guilherme Sperb Machado

Co-Supervision

Ursula Deriu

July 2019

Zusammenfassung

Blockchain-Technologien haben in den letzten Jahren ein überwältigendes Interesse, eine langsam anwachsende Mainstream-Adaption und grosse kommerziellen Investitionen erlebt. Der Einführung neuer anwendungsspezifischer Blockchains folgte das Bedürfnis nach Interoperabilität zwischen den isolierten Technologien. Obwohl breit unterstützt, blieb die Problematik bisher mehrheitlich ungelöst. Dem Austausch von digitalen Vermögenswerten und Währungen zwischen heterogenen Blockchain-Technologien, auch *Atomic Cross-Chain Swaps* genannt, wurde dabei das grösste Interesse zuteil, was zu der Entstehung diverser Crypto-Börsen führte.

Die Mehrheit dieser Börsen agieren jedoch als *Third-Party Custodians*, Verwalter der Private-Keys ihrer Anwender, und benutzen zusätzlich zentralisierte Komponenten für den Wechsel von digitalen Währungen. Technische Schwachstellen, unzureichende Regulationen und grasierende Veruntreuungen von Kundengeldern setzen Anwender daher unnötigen Risiken aus und unterstreichen die Notwendigkeit für dezentralisierte Ansätze.

Mit einem Protokoll namens *Hashed Timelock Contracts* (HTLC) bietet sich eine vollkommen dezentrale Lösung an, welche auf einer lokalen Schlüsselverwaltung und den *Smart Contract* Möglichkeiten der zum Tausch unterliegenden Blockchain-Technologien aufbaut. Dieses Protokoll ermöglicht damit eine finanzielle Privatsphäre und damit Schutz vor repressiven Regimen für ihre Anwender.

Diese Vorteile sind jedoch mit gewissen Kosten verbunden. Dem Paradigmen-Wechsel von einer zentralisierten zu einer dezentralisierten Lösung folgt auch ein Wechsel in der Kontroll-Struktur des Systems. Jeder Agent eines sogenannten *Multi-Agent System* agiert zuhanden einer anderen Autorität und weisst daher ein individuelles Design auf, welches die jeweilige Zielsetzung reflektiert. Um ihre Interessen zu wahren und sich in einem von Unsicherheit geprägten Umfeld koordinieren zu können, wird von einer *Rationalität* der Agenten ausgegangen.

In Absenz einer zentralen Kontrollstelle und in Anbetracht dessen, dass Crypro-Currencies beachtlichen Kursschwankungen unterliegen, kann ein erfolgreicher Währungstausch nicht mehr durch das Protokoll per se garantiert werden, sondern hängt vielmehr von einer adäquaten Insentivierung ab.

Diese Thesis beschäftigt sich daher mit der Struktur und der Insentivierung des HTLC-Protokolls. Im Zuge dessen wird eine *Rational Verification* geführt, basierend auf der Spieltheorie, um das simulierte System-Verhalten mit sozial-wünschenswerten Resultaten vergleichen zu können. Bisher fehlende und notwendige Teile, darunter eine *formelle Definition* des HTLC-Protokolls, das davon abgeleitete *Strategic Game* und eine das Spiel umfassende *Utility Function* werden formuliert.

Die Test-Resultate lassen darauf schliessen, dass Agenten unter gewissen Bedingungen Anreize besitzen, vom Protokoll abzuweichen und dass daher nicht von einer Garantie für einen erfolgreichen Wechsel ausgegangen werden kann. Aus diesen Schlüssen folgend werden die notwendigen Bedingungen für eine erfolgreichen Anwendung des Protokolls diskutiert, sowie mögliche Anwendungszwecke abgeleitet, um die finanzielle Privatsphäre in dieser neuen Ära von Finanztechnologien sicherstellen zu können.

Abstract

Blockchain technologies have seen a meteoric rise in interest ensued by a slowly progressing mainstream adoption and incoming corporate investment. The introduction of new blockchains for specific business-cases in various industries resulted in the need for interoperability between the isolated technologies. Although the effort gained a lot of momentum, interoperability remains a largely unsolved challenge. The exchange of value or assets between heterogeneous blockchain technologies commonly referred to as *atomic cross-chain swap*, arguably sparks the most interest and is today managed by different crypto-exchanges.

The majority of these exchanges though act as *third-party custodians* for their clients' private keys and utilize centralized exchange-capabilities. Technical vulnerabilities, weak regulations, and rampant fraud expose clients to unnecessary risks and highlight the need for innovation.

A protocol called *hashed timelock contracts* (HTLC) offers a fully-decentralized solution, solely based on *smart contract capabilities* provided by most public blockchain technologies and *local custody.* This effectively enables financial privacy and freedom from oppressive regimes for its users.

These advantages do come with a price. The paradigm shift from a centralized to a decentralized solution also comes with a change in the control structure of the system. Each agent in this *multi-agent system* is subject to a different authority and therefore are of different design, reflecting their distinct interests. To be able to ensure their intentions and coordinate accordingly in an environment of mutual distrust, agents are supposed to display *rational behavior.*

In absence of centralized control and since crypto-currencies are subject to remarkable price fluctuations, a successful exchange is arguably not anymore guaranteed by the protocol per se but rather dependant on an adequate incentivization.

Therefore, this thesis considers the structure and incentivization of the HTLC-protocol. A *rational verification* based on *game theory* is conducted in order to compare simulated behavior to desirable system-outcomes. So far missing and necessary pieces, such as a *formal definition* of the HTLC-protocol, the associated *strategic game* and a comprehensive *utility function* are formulated.

The verification under the given models indicates, that agents do have an incentive to deviate from the protocol under certain conditions and that therefore the protocol alone does not guarantee socially-desirable successful exchanges for every case. Subsequently, necessary conditions for successful applicability of the protocol are discussed, as well as possible use-cases derived to enable financial freedom in a new era of financial technologies.

Keywords: Blockchain, Atomic Cross-Chain Swaps, Hashed Timelock Contracts, Multi-Agent Systems, Agent Synthesis, Rational Verification, Decision Making, Game Theory, Expected Utility.

Acknowledgment

I would like to thank my supervisor Dr. Ilir Fetai for his continuous advice on my thesis and his honest feedback. Dr. Guilherme Sperb Machado for his opinions and generally being a great mentor throughout my blockchain-journey. Adrian Burkhalter, for his coaching me the past four years and his invaluable help on so many challenges. Subsequently, I want to thank Marius Giger for being an awesome sport and for all the long hours of discussions we shared. Cornelia and Jürg Rüegger, for their precious bits of advice on my thoughts and for their continuous support during this intensive time. And of course a special thanks to Tabea Murri, for encouraging me with her invincible positivity, for the hours upon hours of conversation, and for always being on my side.

Contents

Chapter 1

Introduction

A new era for financial technologies started with the publication of Satoshi Nakamoto's infamous Bitcoin white-paper in 2008. Bitcoin's peer-to-peer digital cash-system allowed to transfer a scarce digital crypto-currency between unknown parties and removed the need for a trusted financial intermediary with the introduction of the *proof-of-work consensus-protocol.* Blockchain technologies have since seen a tremendous rise in interest with a slowly progressing mainstream adoption and incoming corporate capital[1]. With the appearance of new technologies for specific business cases and the issuance of countless new tokens, the industry deviated from the notion of "one blockchain to rule them all" to a widely accepted perspective of an ecosystem based on numerous heterogeneous solutions [1, pp. 2].

With the increasing usage of digital assets across isolated ledgers, the *interoperability* of distinct blockchain technologies, especially the exchange of digital assets, gained increasing attention. Although having experienced a lot of momentum, interoperability remains a largely unsolved challenge [1, pp. 1-2], with leading blockchain technologies predominantly still operating in almost complete isolation [2, pp. 1].

Quickly *exchanges* for crypto-currencies emerged and started to gain massive traction. As in traditional currency-markets, exchanges provide the necessary *liquidity* for global trade, build bridges between fragmented, heterogeneous ecosystems and reduce the otherwise significant lock-in risk for investors. However, the majority of crypto-exchanges act as *remote third-party custodians* for their clients' private keys and utilize highly centralized solutions for exchange capabilities (i.e. side-chains under full control of these exchanges). As new gatekeepers, these companies not only reintroduce a significant need for trust towards their entities per se but more importantly deny their users the right for financial privacy that blockchain technologies would inherently provide. This is additionally exacerbated by the current vague regulation regarding blockchain-based products [3, pp. 85]. Rampant fraud, multiple allegations of financial misconduct of customers' funds and the exploit of technical vulnerabilities in these centralized systems, raised immense concerns [4, pp. 121] and earned this industry a bad reputation [4, pp. 7][5, pp. 1].

This highlights the necessity for research on new approaches for fully-decentralized autonomous exchange capabilities. Thereby, the arguably most promising approach for a fully-decentralized atomic cross-chain swap is a protocol called *hashed timelock contracts* (HTLC), which leverages *local custody* and is entirely based on the *smart contract* capabilities of the underlying blockchain technologies [6, pp. 1-2].

[1] With total corporate investments over 1.3 Billion Dollars in 2018: https://tinyurl.com/y43ugdnx, last seen at 18.06.2019

1.1 Motivation

Based on decentralized principles, the HTLC-protocol has the potential to enable truly open, neutral, borderless and censorship-resistant financial systems [5, pp. 1] arguably aligned with Satoshi's initial philosophical understanding [2, pp. 1]. Such technology could empower people globally and enable them to take their financial sovereignty into their own hands. Such a system would honor the individual right for privacy and push back the troubling reality of Orwellian states exploiting technological means to control and oppress people of different political shades.

However, it would be foolish to rush out solutions without a proper understanding of their technical basis. This would expose the very data and assets of potentially thousands of users, which these new approaches wanted to protect. It is therefore necessary to examine the nature of these new technological concepts and their implications.

1.2 Goals

Roughly a dozen papers have covered the HTLC-protocol so far. Most notably by Herlihy, who published the most cited and widely incorporated scientific paper on the fully-decentralized approach. In the course of his paper, Herlihy described the protocol and defines multiple theorems. His *third axiom* sparked the author's curiosity:

Herlihy's third axiom	No party has an incentive to break the protocol. [7, pp. 246]

Table 1.1: Herlihy's third axiom

By implication, this means that not a single exchange would fail because of deliberate will. However, the paradigm shift from a centralized to a decentralized solution is followed by a radical change in the control structures of the system. Such as system based on autonomous and independent participants can be interpreted as a so-called *multi-agent system*, where each agent is subject to a different authority. Individual agents, therefore, are distinctively designed to optimize for the respective goals of their agency, as described by Kapitonov [3, pp. 1]. To ensure their interests, they show *rational behavior* to successfully coordinate in an environment of mutual distrust. Additionally, crypto-currencies are subject to heavy price-fluctuations and the HTLC-protocol is known for a rather slow time-to-completion [2, pp. 1].

It is therefore relevant to question, if selfish agents without a centralized control can actually achieve a successful exchange in every single case [1, pp. 11]. If not, inherent system failure could result. This sparked the main question and the associated hypothesis of this thesis:

Question	As a coordination-effort between rational agents and under the assumption of price-fluctuations, does the HTLC-protocol always result in a successful exchange?
Hypothesis	Every single exchange of assets is successfully completed.

Table 1.2: Research interest

To answer this question, a *rational verification* should be conducted, which compares the by Herlihy described as guaranteed desirable outcome to a simulation's result under different price fluctuations. The process should thereby challenge the leading hypothesis deduced from Herlihy's axiom. The hypothesis should be rejected if a single case (a so-called *black swan*) is found. Besides, the results of the verification should be analyzed to find ideal environments for the applicability of the protocol. To stay true with decentralized principles, it should be assumed that only *public blockchains* (e.g. *Bitcoin* or *Ethereum*) are used for exchanging assets. The term *assets* should furthermore exclusively refer to *crypto-currencies* since they have public and transparent market prices available.

1.3 Methodology

Although incentivization had been a major subject of discussions in the blockchain community, especially around the economic stability of the distributed consensus between miners, little effort was effectively put into analyzing incentive structures of *strategic interaction* between parties over distinct blockchain technologies. In his paper, Herlihy missed the opportunity to verify his third axiom. To accept or reject the theorem and thereby answering the main question, it is necessary to analyze the outcomes of exchanges within a multi-agent system using the HTLC-protocol.

Therefore, a method called *rational verification* (also known as *equilibrium checking*) is used, which enables to validate desirable outcomes against a simulation of a given agent's *model of rationality* [8, pp. 4184]. The method is slightly modified to adjust to reflect the testing purposes and is visualized in *Figure 1.1*.

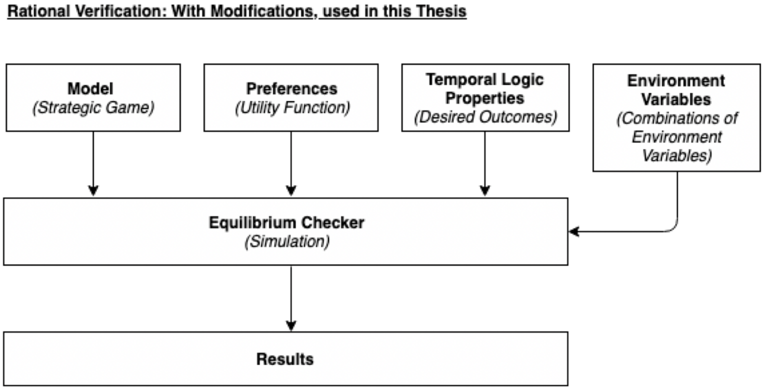

Figure 1.1: Modified rational verification, inspired by [8, pp. 4185]

However, the individual components for the verification are missing, with the exception of the *temporal logic properties* given by Herlihy's third axiom. The missing parts therefore will be incrementally synthesized using the following guiding questions:

Number	Question
i	Which capabilities of blockchain technologies does the protocol utilize?
ii	How is the protocol formally defined?
iii	Which consequences for an agent's design does a multi-agent system imply?
iv	Which model of an agent's rationality can be assumed?
v	Which strategic game can be deduced from the protocol's formal definition?

Table 1.3: Guiding questions

Since the missing components for the rational verification have to be developed first, the structure of the thesis will be aligned with the order of the guiding questions.

1.4 Contribution

This thesis contributes important new insights to the overall scientific discourse surrounding the HTLC-protocol and its *applicability* for atomic-cross chain swaps.

Specifically, a *formal definition* of the protocol is deduced, explaining its workflow on an appropriate level of detail. Additional implementation details and information is provided, giving insight into the time behavior of the said protocol.

The interpretation of the environment as a *multi-agent system* defines a paradigm shift in terms of the decentralization of control. This leads to a shift in agent design, away from benevolent-behaving to rational agents.

For the first time, *agent synthesis* is used to model rationality for the HTLC-protocol using a utility function. The subsequent utilization of game theory to analyze a system's outcomes could trigger future discussions around the applicability of this methodology to general challenges in *blockchain economics*.

To the best of the author's knowledge, this analysis also constitutes an important first-time practice of *rational verification* in the context of blockchain technologies. The conclusions over this thesis' testing could add a relevant piece to the discussion, how agent-based interactions in blockchain technologies could be modeled and investigated prospectively.

Finally, conclusions drawn in this thesis could not only be used as a discussion base for existing solutions but ultimately to track and close severe security vulnerabilities - establishing the HTLC-protocol as a secure decentralized solution for atomic cross-chain swaps and making financial privacy possible for everyone.

1.5 Structure

The thesis is structured according to the guiding questions from *Table 1.3* and therefore follow the subsequent order:

Chapter 2 describes the contextual related work used for this thesis. The main concepts and capabilities of blockchain technologies are subsequently explained in *Chapter 3*. In *Chapter 4*, a formal definition of the HTLC-protocol is deduced. Subsequently, *Chapter 5* introduces the concept of multi-agent systems and their implications on the nature of agent-design. These ideas will then in *Chapter 6* be realized using the concept of agent synthesis, the reasoning about rationality in artificial intelligence. The synthesis results in a utility function, defining an agent's model of rationality. Having all components together, *Chapter 7* introduces the modified version of Wooldridge's rational verification and the exact testing procedure. The verification results are presented in *Chapter 8*. These results are then discussed *Chapter 9* and conclusions about the protocol's applicability in different use-cases are drawn in *Chapter 10*. Finally, *Chapter 11* summarizes the findings, describes plausible fields of applicability, critically reflects the methods used in the process and gives an outlook on future opportunities for research.

Chapter 2

Related Work

A first description of a blockchain technology was given by Satoshi Nakamoto's in his infamous Bitcoin whitepaper, which laid the ground for the succeeding distributed ledger technologies. In his paper, Nakamoto describes the basic principles of a new form of distributed consensus, today known as *proof-of-work* [9]. Buterin and Woods then described the Ethereum blockchain in their white- and yellow-paper, delivering valuable insights into the details of the practical implementation of today's second-biggest blockchain [10][11].

Iansiti and Lakhani gave an outlook on the future impact of blockchain technologies in various industries [4]. Covering especially the financial aspects, Peter et al. defined how banking could be disrupted by smart contract capabilities [12].

Building onto these capabilities, Herlihy published a widely incorporated paper describing the basics of the HTLC-protocol. The research interest of this thesis stems from his theorems, the most important being that no party has an incentive to deviate from the protocol [7]. Buterin [1], Decker [13] and Imoto [14] also analyzed the protocol and added further details for its productive use. Bennick and Gijtenbeek investigated different ways to implement *smart contracts* on Ethereum for atomic swaps. They went into details how reusable or single-use contracts could be designed and shared their blueprints as an outcome [15]. Zamyatin and Gervais expressed their skepticism about the HTLC-protocol by describing several weaknesses, among others the timelocks placed in the protocol under the condition of price fluctuations [2].

Durfee et al. published a paper about multi-agent systems and their implication on the agent design [16]. In their paper, they introduced rational behavior as an agent's key characteristic. The concept of modelling such an agent, referred to as *agent synthesis*, is covered by many, including Wooldridge, Shoham, Kapitonov and Parsons [3][17][18][19][20]. The concept of the *rational verification* was introduced by Wooldridge et al. and serves to verify Nash equilibria within a system [8].

This already impies that Wooldridge's methodology is heavily based on *game theory*. The basic concepts of game theory is explained with an abundance of information in the book of Nisan et al. [21] and completed by the descriptions from Turocy [22]. These explanations have been used for the defintion of the strategic game and the calculations in the rational verification.

Chapter 3

Theory

The chapter provides a theoretical introduction to blockchain-technologies. It thereby presents answers to *Question i*: *Which capabilities of blockchain technologies does the protocol utilize?*

To do so, the chapter illustrates the relevant technological concepts and interpretations of current well-adapted public blockchain technologies. Furthermore, the HTLC-protocol and its applicability should be explained.

3.1 Blockchain

The research on digital currencies and their technological basis had been started a few decades ago, but mostly suffered a niche existence [11, pp. 1-2]. This changed with the publication of Satoshi Nakamoto's famous *Bitcoin* whitepaper in 2008, which incorporated concepts from previous research, such as *smart-contracts* by Szabo or *proof of computational expenditure* by Dwork and Nao [10, pp. 4]. With the realization of the Bitcoin protocol, Nakamoto created a peer-to-peer digital cash system that operated internationally without any central issuer (i.e. usually a national central bank) [10, pp. 1]. By solving the *double-spending problem* (e.g. using the same digital value twice), blockchain technologies effectively enabled *scarcity* of digital assets in decentralized systems [9, pp. 2]. Iansiti therefore describes blockchains as a *foundational technology*, which could radically change how global economic systems will work in the future and could challenge the status quo of traditional financial systems [4, pp. 121].

Blockchain technologies enable mutually distrusting parties to exchange digital assets on a public decentralized ledger [4, pp. 121]. These technologies work without any trusted central intermediary by providing a transparent, permanent and tamper-proof data storage based on cryptographic principles [9, pp. 1] [23, pp. 45]. These cryptographic means and a *consensus algorithm*, such as *proof-of-work*, ensure the irreversibility and replication of newly incorporated transactions across thousands of participating nodes [10, pp. 4][4, pp. 122].

Ownership is implemented using the concept of *private keys.* A private key is a *256-bit number*, governed by the *secp256k1 ECDSA standard*, and is usually stored in some form of a *digital wallet.* Thereby, each private keys controls at least one associated *public key* or *address* and hence, has the ownership over the on this address deposited currencies or digital assets [1]. A *change of ownership* is achieved by using the private key to proof ownership over an asset and subsequently issuing a *transaction* to the a address. Important to note, transactions are issued with a *transaction fee*, later compensating a validating node's computational effort [11, pp. 7].

[1] https://en.bitcoin.it/wiki/Private_key, seen at 24.07.2019

The transaction is then temporarily stored in the so-called *mem-pool*, waiting for validation. A fully-validating node, also referred to as *miner*, subsequently chooses several transactions from the mem-pool, validates and encapsulates them in the eponymous unit of a *block* [10, pp. 4]. Blocks and transactions thereby point to their respective predecessor using a *cryptographic hash function* [23, pp. 45]. This concept creates a data structure of a de-facto *chain of digital signatures* which ensures the integrity of previous information, such that no historical data on the blockchain can be modified without changing the digital signature of the latest block and therefore expose malicious changes [9, pp. 2-3]. Blockchains are therefore characterized as append-only ledgers [24, pp. 245]. This aspect effectively enables the traceability and auditability of data on the blockchain [12, pp. 10].

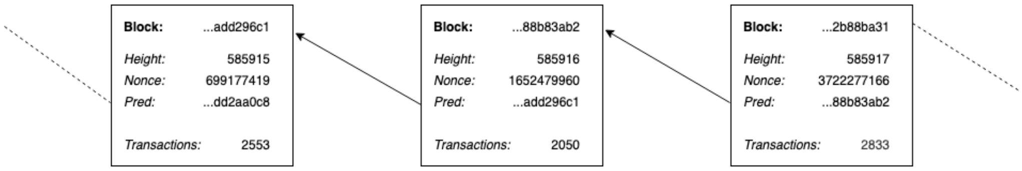

Figure 3.1: Block data structure, e.g. blocks with height 585915-585917

The calculation of the fee depends on several concepts implemented by the underlying blockchain technology. Generally speaking, transaction fee prices emerge on a transparent and competitive market (i.e the mem-pool), where transaction issuers try to get included in a block and miners filling the fixed (and therefore scarce) *block size* [25, pp. 113] with transactions by the *selection criterion* of the highest attached fees per transaction. The competition for validation can therefore result in significant fee fluctuations [2, pp. 14].

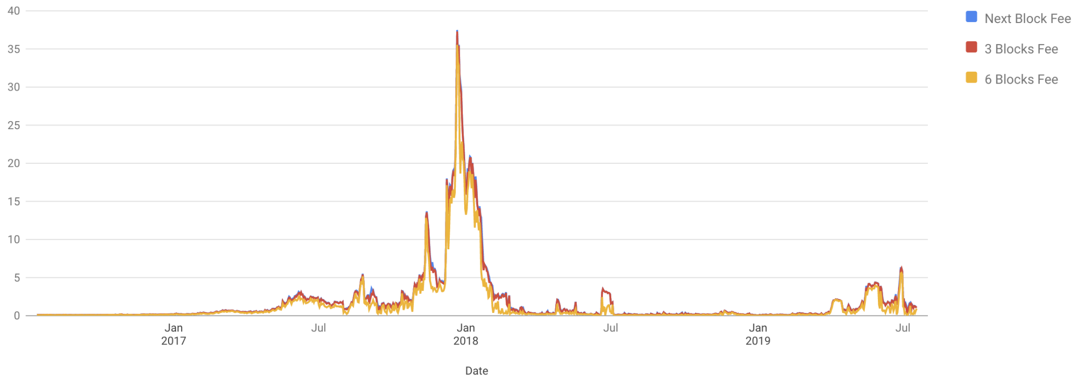

Figure 3.2: Average Bitcoin transaction fees in Dollars from https://bitcoinfees.info (seen at 19.07.2019)

To decide which miner can issue a new block, Bitcoin introduced a new form of decentralized consensus called *proof-of-work*, that closed the gaps of prior research on Byzantine-fault-tolerant multiparty consensus, such as the one used in Adam Back's Hashcash [9, pp. 3]. This form of consensus enabled agreement on the global state of the blockchain in a network of thousands of unknown participating nodes [9, pp. 1][10, pp. 4] and is proven to be productively scalable [25, pp. 112]. Consequently, all miners in the network try to be the fastest in solving a *global cryptographic puzzle*, generally referred to as the *nonce*, to add their block to the existing chain and subsequently broadcast the now *longest chain* to the network [9, pp. 3].

Though, the consensus over the order of the transactions, formerly introduced as a chain of digital signatures, is thereby not absolute but should be rather regarded as a *probabilistic agreement* [25, pp. 112]. There is a chance, that multiple distinct miners find the nonce simultaneously, include different transactions into the new block and therefore create multiple variants of the now longest chain - which effectively results in a *temporary fork* [25, pp. 115].

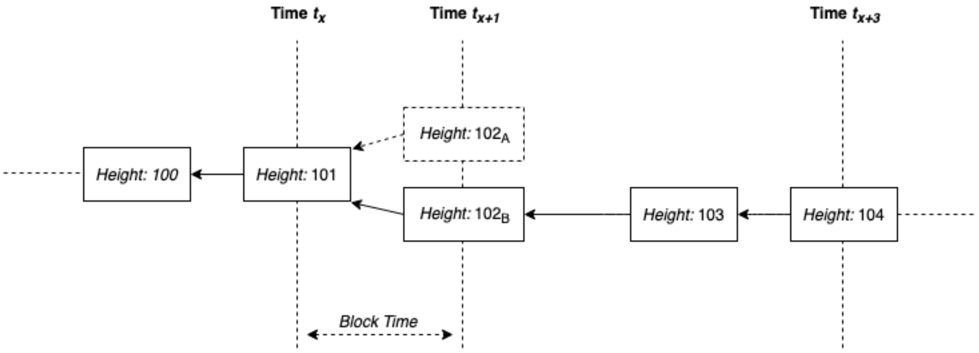

Figure 3.3: Deviating consensus, e.g. an orphaned block

Over time and the inclusion of new blocks, the certainty that an approved transaction is definitely in the chain approximates to a hundred percent and is therefore defined as *probabilistic* [25, pp. 118]. The term *block finality* thereby describes the expected average time to wait for a desired level of confidence and can vary between different technologies depending on their *difficulty* for solving the cryptographic puzzle, and therefore the time for issuing a new block [1, pp. 10][26, pp. 2086].

The resulting stable version of the *longest chain* can be interpreted as the *global state*, which consequentially describes the latest distribution of all digital assets on the respective blockchain. Therefore formally defined, blockchain technologies can be interpreted as a massive distributed and decentralized *state transition system S*, where changes to the global state can be rolled out by issuing a *transaction T* to the network [10, pp. 5]:

$$apply(S, T) \implies S' \tag{3.1}$$

Wood therefore defines a blockchain as: “[...] a cryptographically secure, transaction-based state machine” ([11, pp. 1]). The consensus algorithm thereby ensures and synchronizes the system-wide *replication* of the global state [13, pp. 4]. Although introduced first solely as a peer-to-peer digital payment system, people soon realized that the interpretation of a blockchain as a state machine, its concepts could be utilized for a much broader range of applications. Subsequent technologies such as Ethereum facilitated the deterministic execution of arbitrary code by virtual machines on the decentralized network - so-called *smart contracts* [10, pp. 1].

3.2 Smart Contracts

The concept of *smart contracts* was formerly introduced by Nick Szabo in 1997 and later happened to attract great interest within the blockchain community [12, pp. 7]. Smart contracts are essentially arbitrary digital rules in form of code that have been deployed on a blockchain technology [10, pp. 1] and hence achieve a *consensus on computation* [12, pp. 7]. Because of their self-enforcing nature and (alleged) deterministic behavior, they often get depicted as digital contracts that can be invoked between multiple unknown and distrusting parties - again removing the need to trust a centralized authority [25, pp. 113] (although Peters points out that there is often still the need for a physical and legally-binding contract for real-world applications [12, pp. 7]).

These capabilities allow to build complex and advanced distributed applications, which could involve dozens of smart contracts working jointly to achieve complex tasks [4, pp. 121]. Smart contracts are ideally applicable in a trustless environment due to their transparent and self-enforcing nature - often framed as *"code is law"* [23, pp. 8]. Referring back to the model of blockchains as state machines, these kinds of arbitrary state transitions enabled to implement any kind of use case [10, pp. 1] - for example atomic cross-chain swaps.

3.3 Hashed Timelock Contracts

Over time the industry slowly deviated from a perspective that exclusively Bitcoin as Satoshi's opus will survive and acknowledged that in the future, there will be *specialized* blockchain technologies covering different use cases [12, pp. 2]. This mentality resulted in a rich but fragmented environment of competing technologies, opting for certain strengths and desirable characteristics. Hence, an atomic cross-chain swap is a special case of the overall endeavors for interoperability between the isolated heterogeneous blockchain technologies. As a mechanism that enables multiple parties to exchange any kind of digital assets across different heterogeneous blockchain technologies [14, pp. 1], atomic cross-chain swaps could allow to overcome these technological boundaries and create a comprehensive reciprocally amplifying ecosystem.

Atomic cross-chain swaps can be implemented using different technological approaches which mostly differ in their design choice between *centralized* and *decentralized* elements but usually have a strong aspect in common - they are mostly based on some sort of the formerly introduced mechanism of smart contracts.

This thesis focuses exclusively on the possibly most promising approach for an atomic cross-chain swap - a protocol called *hashed timelock contracts* (HTLC). As a rather raw approach, the protocol was first introduced in 2013 by Tier Nolan in a Bitcoin forum post [1, pp. 10]. Herlihy describes the HTLC-protocol as a special case of a *distributed atomic transaction* [7, pp. 245], whereas the characteristic of *atomicity* is borrowed from the eminent *ACID-properties* known from database transactions [25, pp. 117]. As Buterin states, the protocol as a series of transactions, either succeeds in its entirety or fails completely, and therefore meets the criteria of atomicity [1, pp. 11].

The involved parties exchange value by locally running the asymmetrical protocol using just the involved blockchain technologies' smart-contract capabilities, hence enabling a truly decentralized exchange of value without any intermediaries [14, pp. 1]. Communicating entirely over data published on smart contracts while leveraging local custody, the protocol proposes a *non-custodial decentralized on-chain solution.* Conducted by at least two distributed software components or *agents* representing distinct and mutually distrusting parties, the protocol is conducted in an unknown and unreliable environment without any centralized coordinator.

The HTLC-protocol utilizes contracts with claiming conditions that require the initiating agent to reveal the matching secret to a hash function before a set time-lock expires and hence allows the counter-party to redeem with the now-published secret as well [13, pp. 10]. Those claiming conditions are built as part of a contract function. If a party deviates from the protocol, be it because of technical issues or free will, then no party will end up worse than at the beginning.

To conduct a *rational verification* on the HTLC-protocol, as a primary step its properties and workflow have to be defined in detail. After an unsuccessful best-effort search for such a definition within the existing literature, a formal definition should be deduced in the next chapter.

Chapter 4

Formal Definition

This chapter answer the *Question ii*: *How is the HTLC-protocol formally defined?*

Therefore, the requirements for the execution are stated and the workflow explained in detail. The formal definition thereby loosely follows the annotations from Buterin and Imoto [1, pp. 10][14, pp. 2], but enhances them with additional insights.

4.1 Requirements

The HTLC-protocol between two parties is based on a few assumptions that should be laid out initially. Although possible with multiple parties as shown by Herlihy [7, pp. 246], it should be assumed that only *two parties* are participating in a swap - Alice and Bob. Both parties want to exchange distinct crypto-currencies. Both parties already have the ownership over addresses A_{Ref_i} on the disjoint blockchains B_i with the respective funds they want to trade with. Additionally, both own an address A_{Red_i} on the opposing blockchain $B_{i'}$, where they want to receive funds. Note here, that these addresses do need minimal funding for the redeem transaction. Both parties agreed to a certain initial exchange ratio R_i at time t_0 and a common smart contract code C_i. A leader L was picked between the parties by some kind of choice, in this example it was Alice. Both blockchain technologies have to share a common hashing function $H(s)$, which leader L uses to create a hash h from a prior generated secret s [14, pp. 2]. This hash h is later deployed in both contracts C_i and $C_{i'}$. And since only the redeem transaction from the leader L ultimately reveals the secret s for the hash h, no party can unjustified enrich itself.

$$H(s) = h \tag{4.1}$$

If the involved and distinct blockchain technologies do not share a common hashing function, then obviously hashing a secret would result in a different outcome. It is therefore necessary to first identify possible differences in the blockchain-specific APIs.

All of the discussed information has to be communicated over some form of a communication channel (e.g. an elaborate matching engine or a naive peer-to-peer channel) before the actual exchange.

These assumptions can also be formally defined as a tuple $\Omega = (B, F, P, R, N, L, A_{Ref}, A_{Red}, C, hl, tl)$ whereby the unique variables are defined in *Table 4.1.*

Component	Premise	Implemented
Blockchains: B	$B_i \cup B_{i'} = 1$	$B_A;\ B_B$
Block-Finalities: F	$F_{B_i} > 0$	$F_{B_A} \rightarrow B_A;\ F_{B_B} \rightarrow B_B$
Price of the Currencies: P	$P_i(t) > 0$	$P_{B_A}(t);\ P_{B_B}(t)$
Exchange Rates: R	$R_i(t) > 0$	$R_A = \frac{P_{B_A}(t)}{P_{B_B}(t)};\ R_B = \frac{P_{B_B}(t)}{P_{B_A}(t)}$
Parties: N	$N_i = 2$	$N_A;\ N_B$
Leader: L	$L = 1$	$N_A \rightarrow L$
Refund-Addresses: A_{Ref}	$\forall A_{Ref_i} > 0$	$A_{Ref_A} \rightarrow B_A, N_A;\ A_{Ref_B} \rightarrow B_B, N_B$
Redeem-Addresses: A_{Red}	$\forall A_{Red_i} > 0$	$A_{Red_A} \rightarrow B_A, N_A;\ A_{Red_B} \rightarrow B_B, N_B$
Smart Contracts: C	$C_i \cap C_{i'} = 1$	$C_A \rightarrow B_A, A_{Ref_A};\ C_B \rightarrow B_B, A_{Ref_B}$
Hashlock: hl_{C_i}	$hl_{C_i} = h$, $hl_{C_i} \cap hl_{C_{i'}} = 1$	$hl_{C_A} \rightarrow C_A,\ hl_{C_B} \rightarrow C_B$
Timelock: tl_{C_i}	$tl_{C_i} \cup tl_{C_{i'}} = 1$	$tl_{C_A} \rightarrow C_A,\ tl_{C_B} \rightarrow C_B$

Table 4.1: Formal definition of the HTLC-protocol

The protocol has one major weakness based on the asynchronicity of the underlying blockchain technologies - the slow time-to-completion [2, pp. 1]. This issue is based on the rather limited scalability of blockchain technologies concerning their potential transactions-per-second. Since the finality of transactions and their definitive incorporation into the blockchains global state is probabilistic and based on the underlying blockchain's finality, the HTLC-protocol needs to wait for an appropriate level of probability to trigger the next action. Therefore, the HTLC-protocol does not have a fixed time behavior but rather relies on the underlying blockchain technologies' block finalities F_{B_i} as described in *Chapter 3.1.* These might vary and hence, the individual timely behavior of an exchange has to be calculated individually. This usually results in alternating *time intervals* between the different transactions on the distinct blockchains.

For productive implementations, a term λ should be introduced, which defines a shared time interval as a buffer for parties to adopt - this relaxes the assumption that parties never encounter network lags or other failures and hence can always react immediately. In real scenarios, this assumption would arguably end up in a lot of failed exchanges. Hence, λ can be set to any non-zero value that is suitable for the involved parties and has therefore be communicated beforehand. Also notable is that the *expected time of completion* (ECT) is equal to l_{C_A}.

$$\begin{gathered}
\lambda > 0 \\
tl_{C_B} < tl_{C_A} \quad \text{, whereas } \ \Delta(tl_{C_A}, tl_{C_B}) = \lambda + F_{B_A} \\
tl_{C_B} = F_{B_A} + 2\,F_{B_B} + 3\,\lambda \\
tl_{C_A} = 2\,F_{B_A} + 2\,F_{B_B} + 4\,\lambda \\
ETC = tl_{C_A}
\end{gathered} \tag{4.2}$$

4.2 Procedure

If these requirements are fulfilled and the setup is made, the exchange of currencies can be started. Subsequently, the atomic swap is asynchronously conducted whereby the leader initiates the protocol and the other party follows. A successful case is described in *Table 4.2*.

Action	Party	Description
(1) Initiate	N_A	N_A creates a secret s and a hash-lock h. It creates, signs and publishes a contract on blockchain B_A from its refund address A_{Ref_A} including the hash-lock h, time-lock l_{C_A} and redeem-address A_{Red_B}.
(2) Adapt	N_B	N_B monitors the blockchain B_A for a contract C_A to be mined. Once N_B confirms the newly published contract, it validates the contract's state and code. If both validations meet its expectations, it waits for the blockchain-associated finality F_A to pass. Once passed, it creates, signs and publishes a contract on blockchain B_B from its refund address A_{Ref_B} including the hash-lock h, time-lock l_{C_B} and redeem address A_{Red_A}.
(3) Redeem	N_A	N_A monitors the blockchain B_B for a contract C_B to be mined. Once N_A confirms the newly published contract, it validates the contract's state and code. If both validations meet its expectations, it waits for the blockchain-associated finality F_B to pass. Once passed, it creates, signs and publishes a redeem transaction to the blockchain B_B from its redeem address A_{Red_A} including the secret s to the hash-lock h, before the time-lock l_{C_B} runs out.
(4) Redeem	N_B	N_B monitors the blockchain B_B for a redeem transaction to be mined. Once N_B confirms the newly published redeem transaction, it extracts the secret s. Once extracted, it creates, signs and publishes a transaction to the blockchain B_A, without waiting for the finality F_B to pass, from its redeem address A_{Red_B} including the secret s to the hash-lock h, before the time-lock l_{C_A} runs out.

Table 4.2: Successful exchange

If any party experienced network issues, errors or deviated from the protocol because of other reasons, both parties would be able to refund their respective locked funds with their refund address A_{Ref_i} after the respective time-locks tl_{C_i} have run out [13, pp. 10] - hence no party would be worse off, with the exception of the already paid transaction fees.

However, success is by definition not the only possible outcome. The complete workflow with all options, possible deviations included, is visualized in the following *Figure 4.1*.

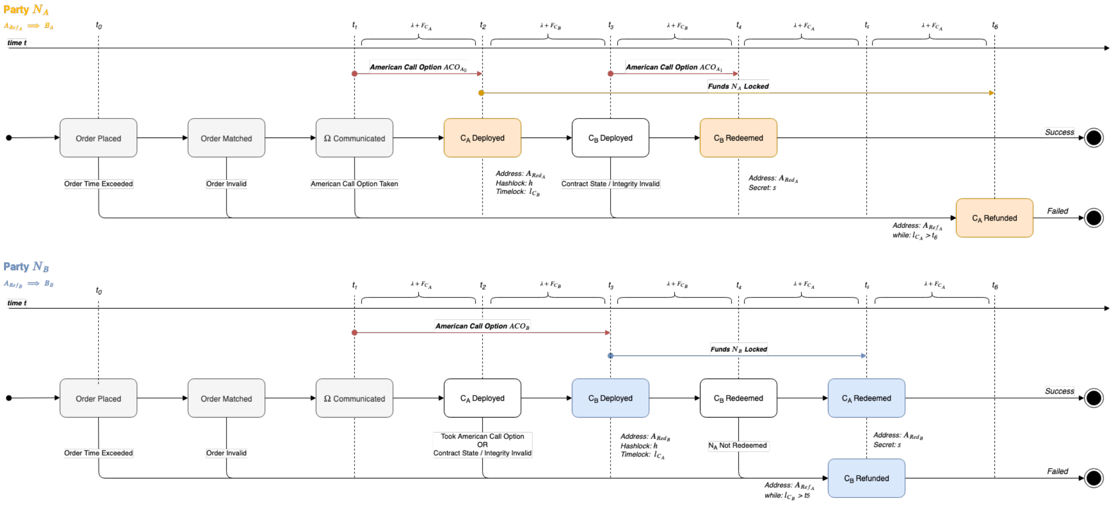

Figure 4.1: Complete workflow

Thereby and as highlighted, both parties have opportunities to *intentionally deviate* from the protocol. The leading party, here N_A, can decide against creating the initial smart contract and later on sending the redeem transaction. The adapting party, here N_B, can only deviate by not creating the adapting smart contract - since it will have no incentive not to redeem once N_A redeemed as well. This nature of decision-making displays a lot of similarities with financial derivatives, especially with *American call options* [6, pp. 4]. Such a call option would give a counter-party the opportunity up to a certain point of time to either activate the option or reject it. However, in contrary to most American call options, these options observed are not priced and hence, open up the opportunity for *arbitrage*.

This observation brings us back to the initial interest of this thesis. As already noted by Buterin and Zamyatin, the slow time of completions in interplay with the exchange rate fluctuations observed on crypto-markets, the socially desirable outcome of a successful swap is not guaranteed and has to be questioned [1, pp. 11].

Chapter 5 therefore provides an introduction to the system, in which the HTLC-protocol takes place and discusses the associated implications.

Chapter 5

System Definition

So far, in *Chapter 3* the relevant technological concepts of public blockchain technologies have been explained and in *Chapter 4* a formal definition for the protocol has been deduced. What has not been defined yet, is the environment in which the HTLC-protocol is executed.

Herlihy defined that: "[an] atomic cross-chain swap is a distributed coordination task where multiple parties exchange assets across multiple blockchains [...]" ([7, p. 1]). This assertion introduces two critical aspects. The system is (1) based on multiple distributed software components, generally called *agents*, which coordinate to (2) accomplish a well-defined task. This definition qualifies the nature of so-called *multi-agent systems* [20, p. 243].

In this chapter therefore, *Question iii* should be answered: *Which consequences for an agent does a multi-agent system imply?*

5.1 Multi-Agent System

Today, the general notion of *agent systems* is widely incorporated in computer science and is part of the broader discipline of artificial intelligence [17, p. 115]. Broadly speaking, agent system research incorporates the efforts of *decision theory* and is interested in the design and analysis of *multi-agent interactions* [20, p. 243-244]. Although research upon agents and their design is getting more visible, the scientific community in artificial intelligence struggles to accept a common definition of an *agent*. However, Wooldridge highlights the central characteristic of *autonomy* for an *agency*: "An agent is a computer system that is situated in some environment, and it is capable of autonomous action in this environment to meet its design objectives" ([18, p. 5]).

So far, an agent system was described as a *distributed environment* still under control of a *single authority*, in which agents share a homogeneous design. Yet, HTLC-based agents exhibit an additional vital characteristic - they are not only distributed, but also *decentralized* [7, p.2]. Ensuing this paradigm change, the individual agents are specifically not controlled by a single entity anymore, but rather by many authorities with heterogeneous interests and targets. Since agents are not anymore orchestrated under a single control structure, they are also expected to be of distinct origin and design, reflecting the diverging needs for their respective agencies [27, p. 74].

This paradigmatic change introduces new challenges for the design of agent systems in general. Durfee et al. proposed to make an active distinction between so-called *multi-agent systems* (MAS) and *distributed problem solving* (DPS) based on a system's control structure and the involved agent's design. As they argued, a multi-agent system distinguishes itself as a system composed of autonomous and *rational* behaving clients pursuing their selfish interests, in contradiction to agents in DPS which qualify themselves by their *benevolent behavior* in achieving a common goal under a system-wide cooperation effort. The interpretation of agent systems therefore moved from systems that "took for granted that agents would be able to agree, share tasks, [and] communicate truthfully [...]" ([16, p. 54]) as the norm, to them being an exception [20, p. 249]. The necessity of autonomous agents to pursue individual goals and coordinate accordingly introduced the notion of *rational agents.*

5.2 Rational Agents

Rationality is a rather vague term that is controversially discussed in computer science [17, p. 118]. As Wooldridge argues, intelligence is closely linked with the *flexibility* in an agent's pursuit of an objective. Hence, he proposes *three common characteristics* that all agents share:

1. Social Ability
2. Reactivity
3. Proactiveness [18, p. 8]

Consequently, (1) agents are able to form a *relevant perception* of their immediate environment, (2) pro-actively plan and taking lead in *objective-directed action* under (3) *communication* with other agents [18, p. 8].

As the definition of interactions for exchanging digital assets, the HTLC-protocol defines *how, with whom and when* to interact [28, p. 2] and therefore sets the framework for an agent's communication. In the case of blockchain-technologies, agents communicate over the data on the blockchain itself [3, p. 85]. Parker therefore defined that such a system is characterized by "interdependencies and feedbacks between the agents and the environment" ([29, p. 317]). However, the protocol itself only defines the rules and the set of interactions, yet fails to give insights about an agent's beliefs or decision-making.

An agent's perception of its environment is often described as a *belief system.* It constitutes the current knowledge about an agent's intermediate surroundings. A belief is therefore highly subjective, based on an agent's available data sources [19, p. 4]. An agent should ideally be capable to identify critical changes in its environment to be capable of immediately react if necessary [30, p. 678]. It therefore might be necessary to bundle and validate data from different sources [30, p. 680]. The importance of a contemporary and relevant belief is stated by Wooldridge, who argues that rationality only exists within the borders of an agent's belief [17, p. 119]. Moore even portrayed *knowledge* as the "precondition for action" [17, p. 128].

In the pursue of analyzing the decision-making of decentralized autonomous agents, the research interest quickly moved to decision theory, especially *game theory* [20, p. 243]. Thereby, game theory as "the mathematical study of interaction among independent, self-interested agents" is widely applied in numerous disciplines ranging from economics, sociology, psychology and evermore as a tool for "automated negotiations" in computer science [31, p. 47]. The application of game theory to these new problems introduced "mathematical techniques", which facilitate to reason about *strategic decision-making* in complex and uncertain environments [20, p. 244, 248-251]. In game theory, decision-making is closely intertwined with the concept of *maximizing welfare* [28, p. 1]. A rational agent is assumed to *strategically* reason for the best alternative over a finite set of possible outcomes, under the consideration of a counter-party's possible decisions and possibly, conflicting interests [20, p. 249].

To analyze preferences over different actions and outcomes, the concept of a *utility function* is commonly applied. The utility function allows to not only to reason about preferences in ordinal terms over possible alternatives, but to quantify an outcome's payoff as a real number [31, p. 47]. However, a utility is always based on an agent's perception of the "state of the world" [20, p. 244-246]. Aggravatingly and as Parsons points out, an agent is "inherently uncertain" about its surroundings [20, p. 244] and therefore bases its calculations on assumptions and estimations.

As Shoham defined it, game theory excels in the "analysis and design of systems that span multiple entities with diverging information and interests" ([19, p. 1]) and is hence perfectly-suited the analysis of a multi-agent system. However, the agent design is not deterministically given but rather underlays a modeling process - generally referred to as *agent synthesis*.

5.3 Agent Synthesis

According to Wooldridge's interpretations, the concept of agent synthesis described the modeling process for intelligent (i.e rational) behavior, which results in *agent theories*. These artificial specifications try to answer the question of how an agent's behavior can be conceptualized [17, p. 119]. The complex reality of (especially human) decision making is thereby subject to abstraction, using methodologies "which provide us with a convenient and familiar way of describing, explaining, and predicting the behavior of complex systems" ([17, p. 121]).

The process of agent synthesis can be interpreted as a "prescriptive practice of game theory" [22, p. 5]. In this process, the designer of a rationality's model is forced to reason about an agent's "knowledge and belief" [19, p. 4], to subsequently formalize a strategic game with associated preferences. It is difficult to capture and reproduce motives in their entirety, as Wooldridge argues [17, p. 117]. Practitioners often face the following problems:

1. The transduction problem
2. The reasoning problem [17, p. 132]

The first issue describes the challenge of translating the complex reality into the abstract symbolism that is software systems. The second issue covers the challenge of relevancy within a certain model and its validity. Wooldridge beautifully illustrated these difficulties by stating: "Even seemingly trivial problems, such as commonsense reasoning, have turned out to be extremely difficult" ([17, p. 133]) because of the highly delicate task of creating meaningful and realistic models.

In a quest for a solution, von Neumann and Morgenstern declared: "What is important is the gradual development of a theory, based on a careful analysis of the ordinary everyday interpretation of economic facts. The theory finally developed must be mathematically rigorous and conceptionally general" ([19, p. 7]).

To the best of the author's knowledge, there had never been an effort to define such an interpretation for the HTLC-protocol. A comprehensive model for an agent's preferences and beliefs should therefore be defined using the methodology of *agent synthesis*.

Chapter 6

Agent Design

In this chapter, an agent synthesis is performed, which results in a specific agent design. Thus, the design at hand answers *Question iv*: *Which model of rationality can be reasonably assumed?*.

As explained in *Chapter 5*, the agent synthesis as a modeling process is the subject to the interpretations of an agent's preferences and belief system. In this process, these interpretations are therefore made transparent and result in (1) a *strategic game*, (2) a comprehensive *utility function* and (3) a *behavior* combining the first two elements.

6.1 Strategic Game

In order to deduce an accurate strategic game, the *game representation* should be defined by analyzing the formal definition from *Figure 4.1*.

As clearly observable, the involved parties alternate in the execution of the different steps of the protocol and issuance of transactions to the blockchain [13, pp. 10]. This clearly implies that a *finite sequential game* in a so-called *extensive form* [21, pp. 66]. The extensive form uses the representation of a decision tree, which reflects the temporal steps in the executions [21, pp. 67][22, pp. 7]. Since both parties can achieve a positive outcome by exchanging assets, a *non-zero-sum game* can be assumed. The protocol defines the possible set of actions, which characterizes a game of *complete information* [21, pp. 67]. And since it is assumed that all parties share a common utility function, which results in transparent strategies, the strategic environment in question can be defined as *perfect information game* [21, pp. 68].

Summarizing, the protocol can be interpreted as a *non-zero-sum finite sequential game under perfect and complete information.*

The described strategic game should be solved by an agent using the *solution concept* of a *subgame perfect Nash equilibrium*, since the game is in the extensive form [21, pp. 68]. Thereby, it is proven that every sequential game under perfect information has a Nash equilibrium [21, pp. 69]. If the structure of the strategic game is known and all payoffs have been defined, the solution can be calculated by applying the *backwards induction* algorithm [21, pp. 69].

Table 6.1 describes the deduced components for the strategic game.

Component	Premise	Denoted by
Agents: N	$N_i = 2$	I, II
Actions: A	$A_i = 6$	$I,\ I',\ A,\ A',\ R,\ R'$
Choice-Nodes: H	$H_i = 3$	$H_1,\ H_2,\ H_3$
Leaf-Nodes: Z	$Z_i = 4$	$Z_1,\ Z_2,\ Z_3,\ Z_4$
Action Function: χ	$\chi = H \to 2^A$	$\chi_1(H_1) \to I, I',\ \chi_2(H_2) \to A, A',$ $\chi_3(H_3) \to R, R'$
Player Function: ρ	$\rho = H \to N$	$\rho_1(H_1, H_3) \to N_1,\ \rho_2(H_2) \to N_2$
Successor Function: σ	$\sigma = H \times A \to H \cup Z$	$\sigma_1(H_1, I') \to Z_1,\ \sigma_2(H_1, I) \to H_2,$ $\sigma_3(H_2, A') \to Z_2,\ \sigma_4(H_2, A) \to H_2,$ $\sigma_5(H_3, R') \to Z_3,\ \sigma_6(H_3, R) \to Z_3$
Utilities: u	$u(t) = Z \to \mathbb{R}$	$H_0(t) * \Big((E(S_t) - S_0) + (S_0 * \gamma) \Big) - \big(H_1(t) * \eta \big)$
Time-Steps: t	$t = H + 1$	$t_0,\ t_1,\ t_2,\ t_3$

Table 6.1: Strategic game specification for the HTLC-protocol

In *Figure 6.1*, the abstract definition is visualized in the *extensive form*, as a *binary decision tree.*

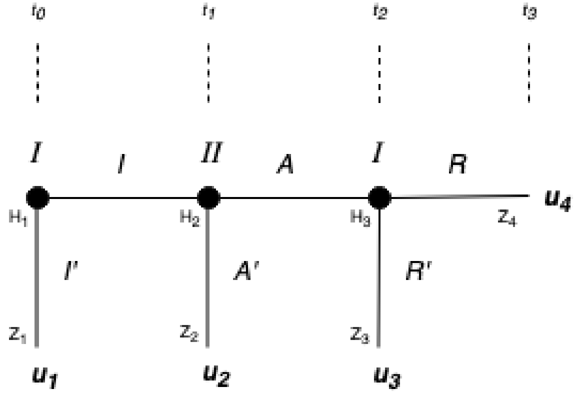

Figure 6.1: The HTLC-protocol as a Strategic Game

It should be noted, that the redeem transaction for Party N_B has been left out in the transition from the workflow visualization in *Figure 4.1* to the strategic game in *Figure 6.1.* It is assumed, that a rational party II would always issue its redeem transaction given that party I already redeemed. For that reason, the party II is associated with only the choice node H_2.

6.2 Utility Function

In order to achieve a generalizable model with high validity, Shoham's advice should be followed: "Rather then start from very strong idealizing assumptions and awkwardly try to back off from them, it may prove more useful and/or accurate to start from assumptions of rather limited reasoning and mutual modeling, and judiciously add those as is appropriate for the situation being modeled" ([19, pp. 6]).

The main objective thereby is not the creation of a perfectly accurate representation of human decision-making, but rather a model congruent with fundamental premises. Over-complex models would run the risk of overfitting the designer's bias and hence potentially deviate from the average behavior or definition of rationality. The utility function defined in this thesis is therefore intentionally of straightforward design.

The following assumptions should be incorporated in the utility function, defining an agent's preferences and reflecting an agent's possible boundaries of belief:

1. A rational agent considers an *expected return* α.
2. A rational agent considers the *subjective significance* of an exchange γ.
3. A rational agent considers *transaction fees* as costs η.

It is thereby realistic to expect an agent to (1) have access to concurrent exchange rate data later used to make simple calculations for an expected return, and (2) being able to monitor and consider transaction fees on the relevant blockchains.

For better understanding, the utility function should be established iteratively, starting with the definition of the *expected return* of an exchange.

6.2.1 Expected Return

Following the research interest, an important part of an agent's belief system is the assumption about future changes in the exchange rate and their effect on the agent's expected returns. The expected value for a digital asset can thereby be calculated using the concept of *geometric Brownian motions* (GBM).

Geometric Brownian motions is an established and often applied *stochastic process* for modelling stock and asset prices in *mathematical finance* [32, pp. 6].

The underlying *Wiener process* is thereby used to create a *random walk* by drawing a random value from a normal distribution $\phi \sim N(0,1)$ [32, pp. 7]. The movements of a trajectory are calculated by a *percentage drift* μ, a *percentage volatility* δ and the Wiener process - as defined in *Equation 6.1* [32, pp. 15].

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma \mathrm{W}_t\right) \tag{6.1}$$

This results in a *Markov process*, which defines that future movements are only dependent on the current situation and but not the past probability distribution [32, pp. 15].

The left part of *Figure 6.2* highlights the nature of the Wiener process as a log-normal distribution by visualizing 2000 random geometric Brownian motions. The right figure displays the distribution of their end-values at time $t_{final} = 100$.

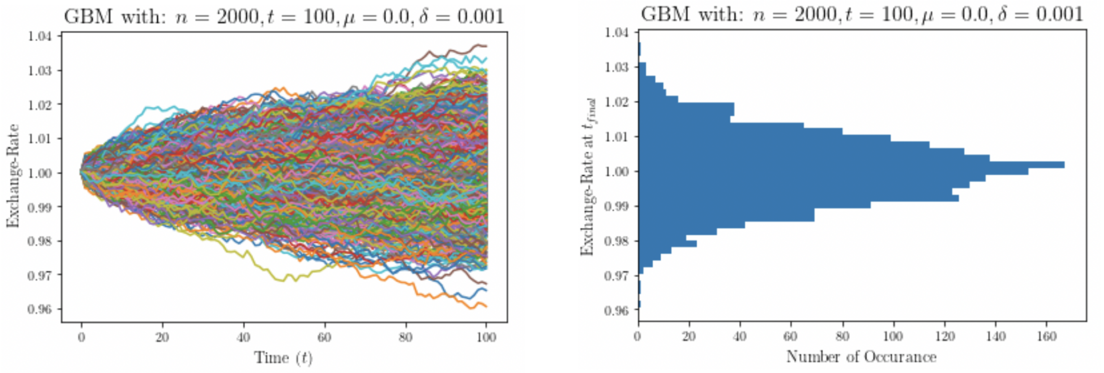

Figure 6.2: Geometric Brownian motions illustration

A geometric Brownian motion's expected value is calculated by *Equation 6.2* [32, pp. 16]:

$$E(S_t) = S_0 \, * \, \mathrm{e}^{\mu t} \tag{6.2}$$

The expected variance is thereby defined by *Equation 6.3* [32, pp. 16]:

$$Var(S_t) = S_0^2 e^{2\delta t}(e^{(\delta^2)t} - 1) \tag{6.3}$$

In order to compute the expected return α, the expected value of an asset as described by *Equation 6.2* is applied in *Equation 6.4*

$$E(S_t) - S_0 \tag{6.4}$$

The exchange return is finalized at the very end of the HTLC-protocol, formally defined by terminal node Z_4 in *Figure 6.1*. The expected payoff should therefore only then be considered. This is feasible with the application of a *Heaviside function* $H_0(t)$, defined in *Equation 6.5*

$$H_0(t_x) = \begin{cases} 1 & \text{if } \ t_x = t_{final} \\ 0 & \text{if } \ t_x < t_{final} \end{cases} \tag{6.5}$$

Adding the findings from *Figures 6.4 & 6.5* together, the expected return α can be calculated as described in *Equation 6.6*.

$$\alpha = H_0(t_x) * (E(S_{t_x}) - S_0) \tag{6.6}$$

6.2.2 Significance

An agent is assumed to always have a reason for exchanging digital currencies, which implies an associated significance γ. The valuation is thereby represented by a percentage of the original exchange rate, as described by *Equation 6.7*.

$$\gamma * S_0 \tag{6.7}$$

Equation 6.8 enhances *Equation 6.6* by the significance γ.

$$H_0(t) * \Big((E(S_t) - S_0) + (S_0 * \gamma) \Big) \tag{6.8}$$

6.2.3 Transaction Fees

As introduced in *Chapter 6.2*, the third part of the utility function is the consideration of costs. These costs occur as transaction fees for the mining effort in proof-of-work networks. Given the heterogeneous blockchain involved in an atomic cross-chain swap, these costs can vary between the distinct technologies (e.g. Bitcoin and Ethereum). To keep the utility function and thereby the complexity of assumptions lean and simple, these transaction fees are represented by a real value *constant* η over any technologies and transaction involved (e.g. thus no differentiation between contract creation and a redeem transaction is made). This is reflected in *Equation 6.9*.

$$\eta_{B_i} = \eta_{B_-i} \tag{6.9}$$

Transaction fees also do not occur in every single step for every involved party, but can rather be described using a *Heaviside function* $H_1(t_x)$, which returns a value for both parties I, II. The function is defined in *Equation 6.10*.

$$H_1(t_x) = \begin{cases} 2, 2 & \text{if } \ t_x = t_{final} \\ 1, 1 & \text{if } \ t_x = t_2 \\ 1, 0 & \text{if } \ t_x = t_1 \\ 0, 0 & \text{if } \ t_x = t_0 \end{cases} \tag{6.10}$$

The consideration of costs in the utility function results in *Equation 6.11*.

$$\eta(t_x) = -\big(H_1(t_x) * \eta \big) \tag{6.11}$$

Combining the acquired individual parts for the consideration of an expected return α from *Equation 6.4*, significance of an exchange γ from *Equation 6.7* and the transaction fees η from *Equation 6.11*, a comprehensive utility function $u(t)$ is defined in from *Equation 6.12*.

$$u(t) = H_0(t) * \Big((E(S_t) - S_0) + (S_0 * \gamma) \Big) - \big(H_1(t) * \eta \big) \tag{6.12}$$

6.3 Behavior

Based on the definition of the strategic game in *Chapter 6.1* and the utility function in *Chapter 6.2*, the agent design is described as goal- and data-driven [17, pp. 135, 139].

To be capable of reasoning about the likelihood of a successful exchange the agent performs the *Algorithm 6.1*.

Algorithm 6.1 Calculate Action Function ρ

```
Input  : choice node: $H_i$
         exchange rate data: $S_t, \mu, t$
Output: action function: $\chi$
player $N_i$, strategic game $T$
monitoring if new choice node $H_i$ equals player function $\rho(N_i)$ then
    calculate strategic game payoffs with utility function
     calculate nash equilibrium with backwards induction on strategic game payoffs

    if nash equilibrium equals terminal node $Z_4$ then
        return action function equals positive
    else
        return action function equals negative
    end
end
```

The adjustment to a up-to-date exchange rate is executed for every choice node H_i that belongs to an agent N_i. As introduced in *Chapter 6.1*, an agent (1) first calculates the payoffs for a strategic game using the utility function and (2) searches for the Nash equilibrium using the backwards induction algorithm. The agent will subsequently cancel the protocol, if the Nash equilibrium is not the final terminal node Z_4.

The simplicity of the design at hand is key for highly specialized and efficient agents, designed to accomplish a single task. As Acre argued, routine tasks only involve little variance and therefore can be solved using the approach each time [17, pp. 135]

Chapter 7

Rational Verification

This chapter proposes the main methodology with which the leading question of this thesis should finally be answered. In this pursuit, a close alternation of a methodology called *rational verification* should be utilized.

7.1 Modification

The process of formal verification as a solution to the *correctness problem* is well-known in the area of computer science as *model checking*, which aims to test for intended programmatic behavior. The system's behavior is thereby tested against an explicit formal specification φ, describing a desirable outcome. φ should therefore be congruent with the system designer's intentions [8, pp. 4185]. How can these principles now be applied to a multi-agent system with rational agents? And how can a system's behavior be verified?

These questions motivated Wooldridge et. al. to introduce a methodology termed *rational verification.* Since the individual agents are homogeneous in their beliefs and interests, correctness should not be defined by the correct execution of an agent's design. The correctness of the system should rather be interpreted as the aggregated result of behavioral patterns from a possibly infinite number of rational agents. The term correctness therefore rather described the final overall states of the system against the predefined desirable outcome φ using the *Nash equilibrium* solution concept. Testing "whether the system will exhibit the behavior φ under the assumption that agents within the system act rationally in pursuit of their preferences/goals" ([8, pp. 4184, 4185]) is therefore specifically called *equilibrium checking.*

The methodology incorporates concepts from the increasingly discussed *algorithmic game theory* - an intersection of game theory and computer science. As illustrated in *Figure 7.1*, this method is based on three steps: the *model*, *preferences* and *temporal logic properties.*

In their original paper, Wooldridge et al. used the method of rational verification under the expectation, that at the beginning of verification, all agents' preferences over a game are already known. This assumption, in reality, appears to be quite strong and has to be rejected for the case at hand. For that reason, slight modifications should be applied to the rational verification methodology:

1. The model is defined as a strategic game. This accurately reflects the actual decision process.
2. Preferences are not fixed but calculated by a utility function and relevant environment variables. This reflects changes in an agent's belief.
3. Environment variables are added and used to simulate different price-fluctuations.

The previous chapters thereby served to introduce the individual missing and necessary pieces necessary to conduct this analysis. The modifications can be tracked in the direct comparison.

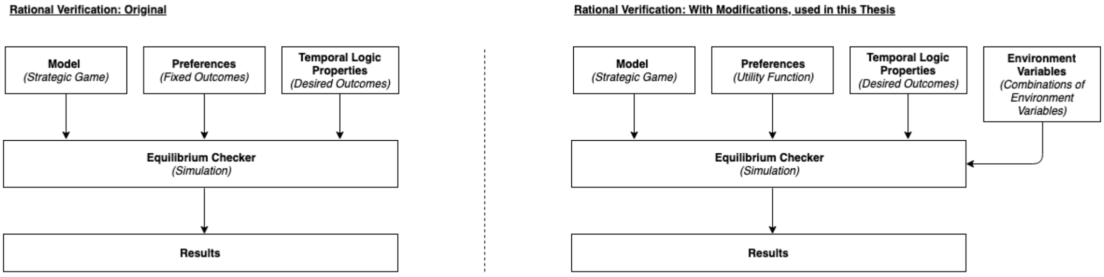

Figure 7.1: Modifications on rational verification, inspired by [8, pp. 4185]

7.2 Setup

To ensure accountable and repeatable testing results, the concrete setup should be described. The following components are therefore used:

- Model: The *strategic game*, introduced in *Table 6.1*
- Preferences: The *utility function*, introduced *Equation 6.12*
- Temporal Logic Properties: Deduced from Herlihy's third axiom in *Table 1.1*
- Environment Variables: Fixed number n of GBM with different drift μ and variance δ

The model of rationality is given by the strategic game and utility function. The goal of the verification is to test, if the decision-process of the involved agents always follows Herlihy's axiom - even under the assumption of price fluctuations. As already explained in *Chapter 6.2* and captured in *Equation 6.1*, price-fluctuations are dependent on the drift μ and variance δ.

The tests to conduct therefore focus on different price fluctuations under increasing μ, δ and are described in *Table 7.1* below.

Focus	Setup
Drift	$t = 100$, $n = 1$, $\mu = 0.001 - 0.003(\Delta = 0.0001)$, $\delta = 0.0$
Volatility	$t = 100$, $n = 100$, $\mu = 0.0$, $\delta = 0.001 - 0.02(\Delta = 0.001)$

Table 7.1: Testing setup

The code is deposited at: https://github.com/unnmdnwb3/rationalverification

Chapter 8

Results

The two predefined tests had been performed according to the description in the previous chapter. Subsequently, the results should be described.

The verification with an alternating range of the drift of a geometric Brownian motion resulted in the following data.

8.1 Drift

In the first test, different trajectories had been plotted by setting the variance δ to 0.0 and shifting the value of the drift μ from $0.001 - 0.003$, with step size Δ of 0.0001.

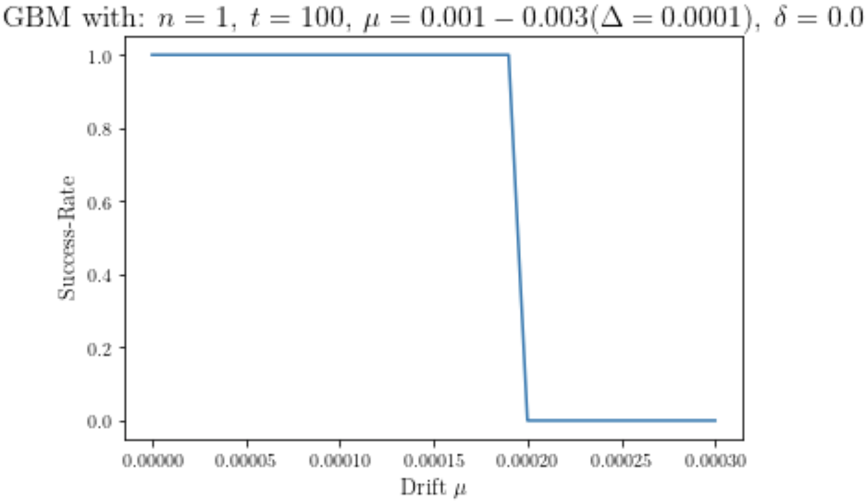

Figure 8.1: Results: testing drift μ

The curve is constant until the test of $\mu = 0.0019$. A clear break is noticeable between the values of 0.0019 and 0.002. Every individual exchange was successful previously and failed after this value. The exact marker though cannot be identified by just interpreting the curve and, hence needs to be discussed in the next chapter.

8.2 Volatility

In the second test, the drift μ was adjusted to 0.0 and the value of the volatility δ was increased along with the range of $0.001 - 0.02$, with step size Δ of 0.001.

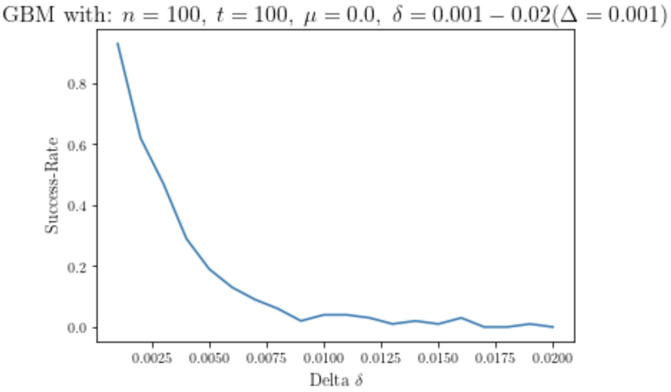

Figure 8.2: Results: testing volatility δ

Variance	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
Success Rate	0.94	0.58	0.37	0.27	0.12	0.09	0.1	0.06	0.03	0.04

Variance	0.011	0.012	0.013	0.014	0.015	0.016	0.017	0.018	0.019	0.02
Success Rate	0.04	0.01	0.03	0.02	0.03	0.0	0.01	0.0	0.01	0.0

Figure 8.3: Results: testing volatility δ

These results reveal a fast decline in the percentage of successful exchanges with increasing variance of δ. Starting from the state $\delta = 0.003$ less than 50 percent and from the marker of $\delta = 0.007$ less than 10 percent of the exchanges are successfully finalized. Also important to note is that even a variance of 0.001 does not result in a guaranteed exchange, but only 94 percent. This might occur since the Wiener process might in some instances result in significant deviations.

Chapter 9

Discussion

This chapter serves to investigate the results presented in chapter 8 and finally to answer the main research interest:

Question	As a coordination-effort between rational agents and under the assumption of price-fluctuations, does the HTLC-protocol always result in a successful exchange?
Hypothesis	Every single exchange of assets is successfully completed.

In the process of rational verification, the results of a simulation with various price-fluctuations was compared to the desired results expressed by the main hypothesis. Thereby, the existence of a single case of a failing exchange would lead to the rejection of the hypothesis and hence, also Herlihy's third axiom.

9.1 Interpretation

For the interpretation for the results, the utility function from *Equation 6.12* should be recalled:

$$u(t) = H_0(t) * \Big(\, (E(S_t) - S_0) + (S_0 * \gamma) \, \Big) - \big(\, H_1(t) * \eta \, \big) \tag{9.1}$$

From this equation, four variables influencing the outcome of a utility function can be identified: drift μ and variance δ defining the geometric Brownian motion used to calculate and simulate the expected exchange-rate $E(S_t)$, an agent's significance of the exchange γ and the value of a transaction fee η. The variables μ and δ had been tested in the subsequent quantitative rational verification.

9.1.1 Drift

The testing of the variable μ clearly showed it's influence on the success of an exchange. As already stated in the results' description, a clear break can be identified after the 0.0019 mark. Under the assumption of near-zero transaction fees $\eta \approx 0$ and the absence of a variance $\delta = 0$, the utility function can be reduced to $(E(S_t) - S_0) + (S_0 * \gamma)$. The term $E(S_t)$ is thereby defined by $S_0 * \mathrm{e}^{\mu t}$. Consequentially, $\mathrm{e}^{\mu t} = \gamma$. The break in the success rate can therefore be identified using the *Theorem 1.*

Theorem 1. *Under the condition δ, $\eta \approx 0$.*

$$successrate = 0, \; if \; \mu > \frac{ln(\gamma)}{t}$$

In the testing procedure, the value of the significance was fixed to a value of 2% or 0.02. This behavior can therefore be replicated when plotting a geometric Brownian motion with drift μ which is calculated using the *Theorem 1.* The result is a value μ of 0.00019802627.

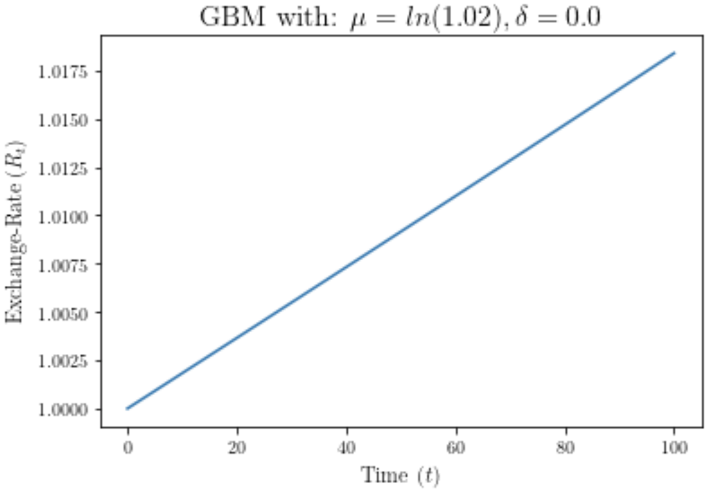

Figure 9.1: Results: ideal drift μ

In general, these results show that the drift of an exchange-rate can influence the outcome of an exchange. An increase in the drift μ has to be followed with an increase in the significance of an exchange γ by the agent. Additionally, these findings indicate that the protocol experiences issues especially when faced with *perfectly negative correlated* assets resulting in a high drift of an exchange rate. In contrast, since exchanging *perfectly positive correlated* assets result in a drift $\mu = 0$, the protocol would always result in successful exchanges under the given assumptions.

Arguably crypto-currencies today are perceived to serve the identical business cases and therefore being pretty much comparable. Thus assumed, that their exchange rates approximate the condition of perfect positive correlation, which by implication would imply that the drift of an exchange rate only has a minor influence of current trades.

However, negative conditions could emerge once digital assets represent real-world commodities and assets. These assets could be diametrical in their price movements, which would increase movements in the respective exchange rates.

9.1.2 Volatility

The results of testing different price fluctuations with increasing volatility δ clearly show that even for small values δ, the success rate drops significantly. The described results show, that even for

$\delta = 0.003$ under 50 percent of all trades are successful. This is congruent with the assumptions deduced from the utility function. As explained in chapter 6, a geometric Brownian motion has a expected variance $Var(S_t) = S_0^2 e^{2\delta t}(e^{(\delta^2)t}-1)$. Since in the testing procedure $S_0 = 1$ and $\mu \approx 0$, the term can be shortened to $Var(S_t) = (e^{(\delta^2)t} - 1)$. In order to find the upper 50-percent confidence interval, the standard derivation has to be calculated using $\sqrt{Var(S_t)} * Z(0.5)$. This results in *Theorem 2*.

Theorem 2. *Under the condition μ, $\eta \approx 0$.*

$$successrate = 0,\ if\ \gamma > Z(0.5) * \sqrt{e^{(\delta^2)t} - 1}$$

Using this definition, a $\delta = 0.003$ would result in $\gamma < Z(0.5) = 0.02074846823$. It is therefore reasonable that only 47 percent of all exchanges can be successfully finalized.

Today, crypto-currency exchange rates are notoriously known for their considerable fluctuations and volatilities. This is especially true for so-called *altcoins*, which represent non-mainstream crypto-currencies often associated with so-called *pump-and-dumbs schemes* where an extensive amount of speculative money is used for a drastic increase of demand, followed by a massive drop in value by immediate sell orders of first wave buyers. Hence, crypto-currencies in stormy waters and altcoins, in general, could be error-prone due to unfavorable and sudden changes in exchange rates.

9.1.3 Transaction Fees

Assumed that an exchange-rate would have a drift $\mu \approx 0$ and a variance $\delta \approx 0$, as a perfect representation of what stable-coins try to achieve in comparison to the pegged currency. This would allow to shorten the utility function to $(S_0 * \gamma) - \eta$. Hence, the *Theorem 3* can be deduced.

Theorem 3. *Under the condition $\mu, \delta \approx 0$.*

$$successrate = 0,\ if\ \eta > \gamma$$

Although seemingly trivial, this shows that even under the assumption of no price-fluctuation, the significance of a trade has to be higher than the involved transaction fees. Since transaction fees are subject to a competitive market, it is not uncommon to see rather strong differences in the number of fees that have to be paid for a transaction to be included. Especially in bull-markets, transaction fees in Bitcoin and Ethereum might rise significantly when mining efforts can not successfully satisfy customer demand in terms of waiting transactions in the mem-pool. Small to medium-sized swaps might not be completed since transaction fees exceed one party's significance of the trade. This factor can be neglected when exchanging high values with a thereby higher value for the significance of η.

9.1.4 Transparency

All of these interpretations are based on a critical assumption: the prevalent transparency of a market with a relevant number of market participants. This would explicitly not be the case in the so-called *over-the-counter trade* (OTC). In this form of non-transparent markets, exotic assets or derivatives get traded by a small number of participants. This results in the absence of public market-prices and hence increases the applicability of the HTLC-protocol dramatically. Since no exchange-rates are accessible, no party has incentives to deviate from the protocol because of exchange rate changes.

9.2 Applicability

Given the interpretations of the results from the rational verification, the hypothesis of this thesis has to be rejected.

✗	Every single exchange of assets is successfully completed.

Table 9.1: Hypothesis check

It is indicated that the HTLC-protocol might not result in positive outcomes under price-fluctuations as well as high transaction fees. The protocol is therefore not be assumed to be universally applicable but should be only implemented given specific conditions. The following table should therefore give advice for which use-case the applicability of the protocol is either given or not.

Advice	Use-Case	Reasoning
(✓)	Established Currencies	Volatility heavily dependable on the current conditions.
(✗)	Altcoins	Associated with a high drift and even bigger volatility.
✓	Stablecoins	Inherent near-zero drift and volatility.
✓	OTC	No transparency, hence no public exchange rate.

Table 9.2: Applicability of the HTLC-protocol under different use-cases

Advice	Condition	Reasoning
✓	Pos.-corr. Currencies	Given near-zero volatility, minimizes exchange rate drift to near-zero.
✗	Neg.-corr. Currencies	Maximizes exchange rate drift, hence minimizes the probability of a successful exchange.
✗	High Volatility	High volatility results in low success rates.
✓	Low Volatility	Low volatility results in high success rates.
(✓)	Bear Market	Less transactions and therefore minimal transaction fees.
(✗)	Bull Market	More transactions and therefore maximal transaction fees.

Table 9.3: Applicability of the HTLC-protocol under different conditions

If one of the potentially negative conditions emerges, the likelihood of a successful exchange decreases. As already stated by Buterin, the existence of block-finalities proves to be challenging for the applicability of the protocol. The implemented consensus algorithm directly influences the speed with transactions included are and therefore also the time-to-completion of the protocol [1, pp. 10]. Because of geometric Brownian motions' exponential nature, blockchain technologies with a relatively fast block finality are therefore preferable.

Chapter 10

Conclusion

This thesis provided insights into the applicability of the HTLC-protocol under the condition of price fluctuations. It thereby challenged the current interpretation, that the protocol guarantees a successful trade. The results at hand indicate that the hypothesis of such a guarantee has to be rejected and that price fluctuations do have an immediate influence on an exchange's outcome, rational behavior on behalf of the involved agents assumed. This is a pivotal insight concerning the applicability of the protocol.

This conclusion is directly based on the applied method of rational verification and therefore also its components. These individual pieces had been synthesized in the process of this thesis and are therefore linked to the interpretations of the author. The significance of this thesis' statements therefore relies directly on their validity.

Game theory and especially expected utility theory had been criticized in the past as an abstract concept, which de-facto represents simplified assumptions about an agent's knowledge and intentions. Especially talking about economic interactions, research has shown that this theory often fails to explain complex human decision making. It could be argued that models of *bounded rationality* would better reflect human decision-making. Hence, it is not entirely coincidental, that this concept was subject to raising interest [29, pp. 325]. However, essays on the perfect representation of human decision-making could fill libraries. For the research interest at hand, it is fair to assume that utility functions perfectly suit the needs for the modeling of rationality.

Since to the best of the author's knowledge, there is no data available on the outcomes and respective environment variables for an HTLC-protocol-based system, a utility function had to be deduced. The definition of the utility function itself was created in the process of the agent synthesis and is therefore based on interpretation and perceptions of the author. To ensure maximal significance, the model was kept as simple as possible - closely aligned with Rosenschein's advice. Covering only the basic considerations, that an agent might go through, the resulting utility function should be able to give valid insight into the decision-process of rationally behaving agents.

One might criticize the to some degree paradox usage of a homogeneous utility function shared by all agents, while at the same time describing the heterogeneity of agents as a critical assumption of multi-agent systems [29, pp. 336]. However, the goal of the thesis was explicitly to question success guarantees under the conditions of a rationality model and price fluctuations. The greater implications of these conditions should be reasonably transferable, also to systems with heterogeneous and more complex representations of rationality.

Also the chosen agent design is rather simple. Although allowing for dynamic behavior and flexibility, agents do not exhibit evolutionary learning capabilities and therefore cannot learn from previous interactions. However, since agents in atomic cross-chain swaps are identified by their public address [3, pp. 84] and hierarchical deterministic wallets are widely incorporated, they possess an infinite number of pseudonyms. This negates any possibility of punishing a malevolent counter-party in the future. Arguably, the lack of persecution even more strengthens the assumption of intelligent and rational agents [28, pp. 2].

The prominent use of geometric Brownian motions could also be questioned since they do not completely represent realistic stock behavior. A static drift and volatility, as well as the impossibility of representing unexpected events in the form of sudden shifts, resulting in a somewhat abstract behavior. Although this is technically correct, great differences in drift and variance would anyway not be anticipated in such a short simulation. Additionally, it was not the intention to perfectly model these price fluctuations, but rather analyze the greater impact of their occurrence.

In summary, given the assumption that the results and conclusions drawn in this thesis are indeed valid and therefore changes in interpretation and application of the protocol is required - why have such behavioral patterns not been witnessed to be widely incorporated by companies and researchers?

Arguably, a lot of papers adapted Herlihy's axioms and hence, incorporated the notion that under no circumstances a rational party has any incentive to deviate from the protocol. If researchers and software engineers believe in these axioms and do not question Herlihy's axioms, paradoxically systems where a naive party has no incentive to deviate, could emerge. Admittedly, such an implementation could be successful in the short run if everyone ignores the obvious. However, once rational agents enter these markets, they could achieve tremendous amounts of arbitrage on the back of naive agents. This could quickly result in a massive downfall of trust towards these executive exchanges and even in a systematic failure of such a system.

As already pointed out by Rosenschein, the fallacy of assuming that decentralized autonomous agents share non-conflicting objectives seems to be a common phenomenon [33, pp. 227]. In reality, agents rarely have identical interests, but are often willing to cooperate in "mutually beneficial activity" to achieve their objectives [33, pp. 228]. Under the absence of benevolent behavior and the acknowledgment of a decentralized control of power though, there is a need for a proper incentive-structure facilitating cooperation among rational agents. Only by recognizing this fact, a stable and tamper-proof system can be established [22, pp. 5]. This security issue has therefore consequentially to be fixed before malicious parties can exploit it - and the finding of this thesis could assist and encourage professionals in the effort of doing that.

Chapter 11

Future Work

With the conclusion drawn in this thesis, there is a variety of opportunities to do further research on.

The results from the rational verification gave insights how such a multi-agent-system might behave, given the protocol and rationality. These results have to be tested empirically, ideally in a productive environment. This could result in new findings about actual agent behavior and exchange-wide patterns that then again could be reflected in agent synthesis.

This thesis aimed to give a rough understanding of the nature of such an agent synthesis and the inherent challenges. However, the proposed agent design is only one interpretation and is therefore open for discussion. The descriptions provided by Wooldridge demonstrate the highly diverse possibilities to design an agent [17, pp. 18-27]. With the emergence of new agents and therefore different models of rationality, distinct strategies could be observed which result in totally new equilibria.

Thereby, the HTLC-protocol itself is also far away from being perfect. The slow time-to-completion, potentially high transaction fees and it's dependency from price-fluctuations could hinder broad usage [2, pp. 1-2]. In agreement with Buterin's statement that "[...] in the case of public blockchains particularly it is impossible to avoid discussing economics" ([1, ppp. 15]), the result of the rational verification highlight the need to investigate how the incentive-structure of the protocol could be adequately adjusted.

Different solutions have so far been proposed. Zamyatin and Heilmann suggest to apply *escrows* or *collaterals* [2, pp. 5-6][6, pp. 3]. The proposed solutions though introduce again centralized components, since otherwise the protocol would have to carry out at least additional two transactions. The general idea of implementing *option pricing* (e.g. under the *Black-Scholes formula*) is comprehensible, since the concept applied million-fold per day on international stock markets [32, pp. 19]. In addition, it would reflect the described behavior of the protocol to create American call options.

However, the protocol might even need more radical changes in its incentive-structure. The utilization of *mechanism design*, a rapidly growing branch of algorithmic game theory described by Roughgarden and Shoham, might be a promising approach to analyze and design the incentivization from scratch. This methodology has proven itself to be exceptionally useful in its application for digital auctions [19, pp. 3].

In conclusion, the HTLC-protocol is a rather new technology for atomic cross-chain swaps, which could enable financial privacy and sovereignty for it's users. However, the conclusion drawn in this thesis indicate, that the protocol is not suited for every asset exchange and might result in undesirable outcomes, given certain conditions. The proposed fields of research though could eliminate the experienced shortcomings and could eventually establish a universal applicability across all use-cases and the support of all kinds of emerging digital assets.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] V. Buterin, “Chain Interoperability,” *R3 Research Paper*, 2016.

[2] A. Zamyatin, D. Harz, J. Lind, P. Panayitou, A. Gervais, and W. Knottenbelt, “XCLAIM A Framework for Blockchain Interoperability,” in *40th IEEE Symposium on Security and Privacy*, no. July 2018, 2018. [Online]. Available: https://eprint.iacr.org/2018/643.pdf

[3] A. Kapitonov, S. Lonshakov, A. Krupenkin, and I. Berman, “Blockchain-based protocol of autonomous business activity for multi-agent systems consisting of UAVs Blockchain-based protocol of autonomous business activity for multi-agent systems consisting of UAVs,” in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, 2017, pp. 84–89.

[4] M. Iansiti and K. R. Lakhani, “The Truth About Blockchain,” *Harvard Business Review*, vol. 95, no. 1, pp. 118–127, 2017.

[5] M. Black, T. Liu, and T. Cai, “Atomic Loans: Cryptocurrency Debt Instruments,” pp. 1–13, 2019. [Online]. Available: http://arxiv.org/abs/1901.05117

[6] E. Heilman, S. Lipmann, and S. Goldberg, “The Arwen Trading Protocols,” 2019. [Online]. Available: https://arwen.io/whitepaper.pdf

[7] M. Herlihy, “Atomic Cross-Chain Swaps,” in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, 2018, pp. 245–254. [Online]. Available: http://arxiv.org/abs/1801.09515

[8] M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, and A. Toumi, “Rational Verification: From Model Checking to Equilibrium Checking,” *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, pp. 4184–4190, 2016. [Online]. Available: http://ora.ox.ac.uk/objects/uuid:1d382e2d-e42c-43de-84b8-c2ceb8ee12f2

[9] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system.” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] V. Buterin, “Ethereum Whitepaper - A Next Generation Smart Contract & Decentralized Application Platform,” 2014.

[11] G. Wood and V. Buterin, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” *Ethereum Project Yellow Paper*, 2014. [Online]. Available: http://www.cryptopapers.net/papers/ethereum-yellowpaper.pdf

[12] G. W. Peters, E. Panayi, and C. Science, “Understanding Modern Banking Ledgers through Blockchain Technologies : Future of Transaction Processing and Smart Contracts on the Internet of Money,” *Banking beyond banks and money*, pp. 239–278, 2015.

[13] C. Decker and R. Wattenhofer, “A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels,” in *Symposium on Self-Stabilizing Systems*, 2015, pp. 3–18.

[14] S. Imoto, Y. Sudo, H. Kakugawa, and T. Masuzawa, “Atomic Cross-Chain Swaps with Improved Space and Time,” pp. 1–17, 2019.

[15] P. Bennik and L. van Gijtenberg, “A Analysis of Atomic Swaps on and between Ethereum Blockchains using Smart Contracts,” 2018.

[16] E. Durfee and J. Rosenschein, “Distributed problem solving and multi-agent systems: Comparisons and examples,” in *Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop*, 1994, pp. 52–62. [Online]. Available: http://www.aaai.org/Papers/Workshops/1994/WS-94-02/WS94-02-004.pdf

[17] M. Wooldridge and N. R. Jennings, “Intelligent Agents: Theory and Practice,” *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[18] M. Wooldridge, “Intelligent Agents: The Key Concepts,” in *ACAI 2001: Multi-Agent Systems and Applications II*, 2001, pp. 3–43.

[19] Y. Shoham, “Computer Science and Game Theory,” *Commun. ACM*, pp. 1–10, 2008.

[20] S. Parsons and M. Wooldridge, “Game Theory and Decision Theory in Multi-Agent Systems,” *Autonomous Agents and Multi-Agent Systems*, pp. 243–254, 2002.

[21] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory.* Cambridge University Press, 2007.

[22] T. L. Turocy and B. V. Stengel, “Game Theory-CDAM Research Report LSE-CDAM-2001-09.” *Centre for Discrete and Applicable Mathematics, London School of Economics & Political Science*, pp. 1–39, 2001.

[23] K. Wüst and A. Gervais, “Do you need a Blockchain?” in *Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2017, pp. 45–54.

[24] M. Herlihy and B. Liskov, “Cross-chain Deals and Adversarial Commerce,” 2019.

[25] M. Vukolić, T. Quest, B. Fabric, and P.-o.-w. Bft, “The Quest for Scalable Blockchain Fabric : Proof-of-Work vs . BFT Replication Marko Vukolić To cite this version : HAL Id : hal-01445797 The Quest for Scalable Blockchain Fabric :,” 2017.

[26] F. Tschorsch and B. Scheuermann, “Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies,” 2016.

[27] J. Paulin, A. Calinescu, and M. Wooldridge, “Agent-based Modeling for Complex Financial Systems,” *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 74–82, 2018.

[28] S. D. Ramchurn, D. Huynh, and N. R. Jennings, “Trust in multi-agent systems,” *The Knowledge Engineering Review*, vol. 19, no. 1, pp. 1–25, 2004.

[29] D. C. Parker, S. M. Manson, M. A. Janssen, M. J. Hoffmann, P. Deadman, S. M. Manson, M. A. Janssen, M. J. Hoffmann, and S. Hall, “Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change : A Review,” *Annals of the association of American Geographers*, vol. 93, no. 2, pp. 314–337, 2002.

[30] M. P. Georgeff, A. L. Lansky, and M. Park, “Reactive Reasoning and Planning,” *AAAI*, vol. 87, pp. 677–682, 1987.

[31] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, 2010.

[32] P. A. Forsyth, “An Introduction to Computational Finance Without Agonizing Pain,” *School of Computer Science, University of Waterloo*, 2017.

[33] J. S. Rosenschein and M. R. Genesereth, “Deals Among Rational Agents,” *Readings in Distributed Artificial Intelligence*, pp. 227–234, 1985.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich diese Thesis selbständig verfasst und keine andern als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche kenntlich gemacht. Ich versichere zudem, dass ich bisher noch keine wissenschaftliche Arbeit mit gleichem oder ähnlichem Inhalt an der Fernfachhochschule Schweiz oder an einer anderen Hochschule eingereicht habe. Mir ist bekannt, dass andernfalls die Fernfachhochschule Schweiz zum Entzug des aufgrund dieser Thesis verliehenen Titels berechtigt ist.

Ort, Datum, Unterschrift

FFHS

Fernfachhochschule Schweiz
Zürich | Basel | Bern | Brig

Mitglied der SUPSI

Departement Informatik
Dr. Oliver Kamin

Title of work:

Hashed Timelock Contracts: An Incentive Analysis

Agent Synthesis and Rational Verification for fully-decentralized Atomic Cross-Chain Swaps

Thesis type and date:

Bachelor's Thesis, July 2019

Supervision:

Dr. Ilir Fetai & Dr. Guilherme Sperb Machado

Co-Supervision:

Ursula Deriu

Student:

Name:	Janick Rüegger
E-mail:	janick.rueegger@gmail.com
Legi-Nr.:	12-749-305
Semester:	FS 2019